Master thesis on Sound and Music Computing

Universitat Pompeu Fabra

Real-Time Multi-Track Mixing For Live Performance

Joaquín Jiménez-Sauma

Supervisor: Frederic Font

August 2018

Table of Contents

1. Abstract

Finding the balance of sounds in a multitrack recording is always a time consuming process performed by experienced professionals. A poor mix produces a mix where is difficult to make the sounds stand out or enhance their presence, also creating a recording where clarity is not perceived. Auditory Masking of tracks is a common problem affecting the presence of instruments in the mix, making some elements indistinguishable and not audible.

This thesis analyses previous research in the field of automatic mixing to find methods to avoid Auditory Masking in multitrack performance. A set of tools are developed during this process to help reduce Auditory Masking by implementing different state-of-the-art techniques as well as a standard measurement of Auditory Masking.

Previous research in the field of automatic mixing is intended to improve mixing for studio recordings. This thesis aims to apply this knowledge to the case of real-time performance where different considerations apply. Unlike similar approaches in the state of the art, the unmasking tool implemented during this thesis is designed to work in real-time.

Three different versions of unmasking tools between two channels in real-time are developed as a result of this research. Each version implements different techniques with similar objectives but different advantages. The result of using this tools is evaluated both quantitatively and quantitatively to find which one provides the best results.

Using tools like the ones developed in this thesis have the potential of helping both experienced and amateur music producers and performers.

2. Introduction

2.1 Motivation

Auditory Masking in music is a common problem in real-time performance where a performer needs the assistance of a live mixing engineer to help the performer keep the clarity and presence of all sounds involved in the musical performance.

In the field of electronic musicians who perform live, the set up is in most cases, composed of small, or portable hardware and/or a computer. This configuration or resources available don’t always allows the inclusion of an audio engineer in the performance. Mixing of multitrack music is also an artistic process. Some artists prefer to keep their mixing decisions saved as part of their tool, and use them every time they perform.

This is were we need to separate the technical side of the performance setup from the artistic and creative tasks. There is a great opportunity and challenge to create production tools that help the performer concentrate on their creativity. In this thesis, in particular, I propose to explore the creation of mixing tools to help the performer unmask, and in turn gain, clarity in the mix of sounds in the production.

The areas of Intelligent Music Production and Intelligent Mixing are quite new, and even when a lot of progress has been made on this area of research, and many solutions for mixing a multitrack recording have been proposed for studio sessions, the case for a real-time tool is not clear yet. No definitive tools, or commercial apps are available for real-time performers at the time of writing this thesis.

The importance of a great mix in real-time performance lies in the fact that it is one of the last steps in the audio processing chain. Mastering is the last step in the chain, but real-time mastering is a complex task, and in the case of an automated tool, it requires a high amount of knowledge, training, prediction and intelligence.

2.2 Research Question

In this thesis we focus on the particular issue of Auditory Masking, and its possible solutions using an intelligent tool. Can we create a reliable system to help a live performer of real-time computer music reduce the amount of auditory masking in a multitrack mix so he can focus on the creative aspect of his performance? What are the obstacles, benefits or disadvantages of trying to create tool for auditory unmasking of real-time audio?

2.3 Objectives

In the process of implementing a real-time mixing tool I am doing a study of previous research on this subject and I will use this results as a starting point for the development of a reliable tool to help unmask one or more channels in a multitrack mix.

The results will be evaluated considering its relevance in the particular case of a real-time tool, and will be implemented as Max for Live patches. This tools focuses on electronic musicians playing mostly electronic music using Ableton Live.

The particular objectives during this research will be oriented on processing at a considerable amount of CPU time (to avoid limiting the performer in the use of his tools), a reduced amount of latency, and ease of use. The latter follows the idea of having a tool that lets the performer focus on creativity more than technical aspects of audio, this tool needs to present a reduced number of controls to stop the performer to worry about the settings of the unmasking tools.

2.4 Structure Of This Thesis

Introduction
Presentation of the research question, motivation, scope and objectives of this thesis.

Essential Theory
This chapter goes through the important theory that is the basis for the implementation.

State Of The Art
This section describes previous research, and shows the relevant works used as the basis of this thesis, describing the reasons to use each one and the way it will contribute to the tools developed in this work.

Proposed Tool

I present in this chapter a discussion and definitions about the challenges and strategies for development of this tool, the differences between versions and research used for each one. As well as restrictions in its use and an introduction to the development tool. A walkthrough of the most important aspects of the development and strategies used to achieve the objectives of this tool.

Using the tool

It is necessary to know what the tool does and how to interpret the results

Evaluation

In this chapter I present the methodology to evaluate the results of using the three tools developed during this research.

Conclusions and Future Work

This chapter sums up the work and experiments done during this research and evaluates the results. Also, comment on the work that I can do to improve this research, directions it may take and possible spin-offs.

3 Essential Theory

3.1 Multitrack Mixing

Multitrack Mixing occurs after recording of separate channels of a musical piece, typically one instrument per channel. This process is commonly performed in front of a mixing desk or in a Digital Audio Workstation (DAW) inside a computer. It aims to balance the sounds in a recording, representing an optimisation problem where there is not a single right answer and many goals may be the purpose or end state. The goals range from adapting the recording for a specific audience, performance location, live performance or to bring emotional impact. Mixing can also be a performance act, as the kind of mixing manipulation during tape concerts, electroacoustic concerts or dub mixing.

Mixing engineers know how to perform the mix of a multitrack recording, they know how to create great mixes, but they might not know how to explain them [19], this means, there is not a strict set of rules to follow when creating the mix of a multitrack recording.

Typically, the multitrack signal goes into a series of audio processes like gain, panning, equalisation, dynamic range and time domain processing. This process may not include all the steps mentioned, neither in the same order, it all depends on the mixing engineer decisions [22].

3.2 Artistic Mixing vs. Technical Mixing

A mixing engineer can create a mental mix of the recording just by listening to the raw multitrack if he is experienced enough. This mix is the engineer’s vision and is likely to change thanks to the input from the producer and/or artist [20]. The same engineer can create a mix making all instruments sound clearly, balanced and separately without disappearing due to auditory masking. Some mixing engineers think of this task as a performance once they achieve some basic balance settings [21]. This is what separates the artistic from the technical mixing.

Basic balance settings means a state of the mix when the engineer can start adding character or imprinting characteristics of specific goal in the final mix.

This thesis will deal with the task of technical mixing, creating tools to be used as an assistant in the process of mixing, allowing the mixing engineer to focus on the creative aspect of mixing after achieving some basic balance settings.

3.3 Frequency Masking

Frequency Masking is a phenomenon in which the perceived audibility of one sound is affected by the presence of another sound which has a similar spectral content [3]. The sounds in a mix compete with each other in multitrack recordings. The result of masking is lack of clarity.

Figure 1 shows an example of a masker and masked sound. The masker can mask many sounds at once.

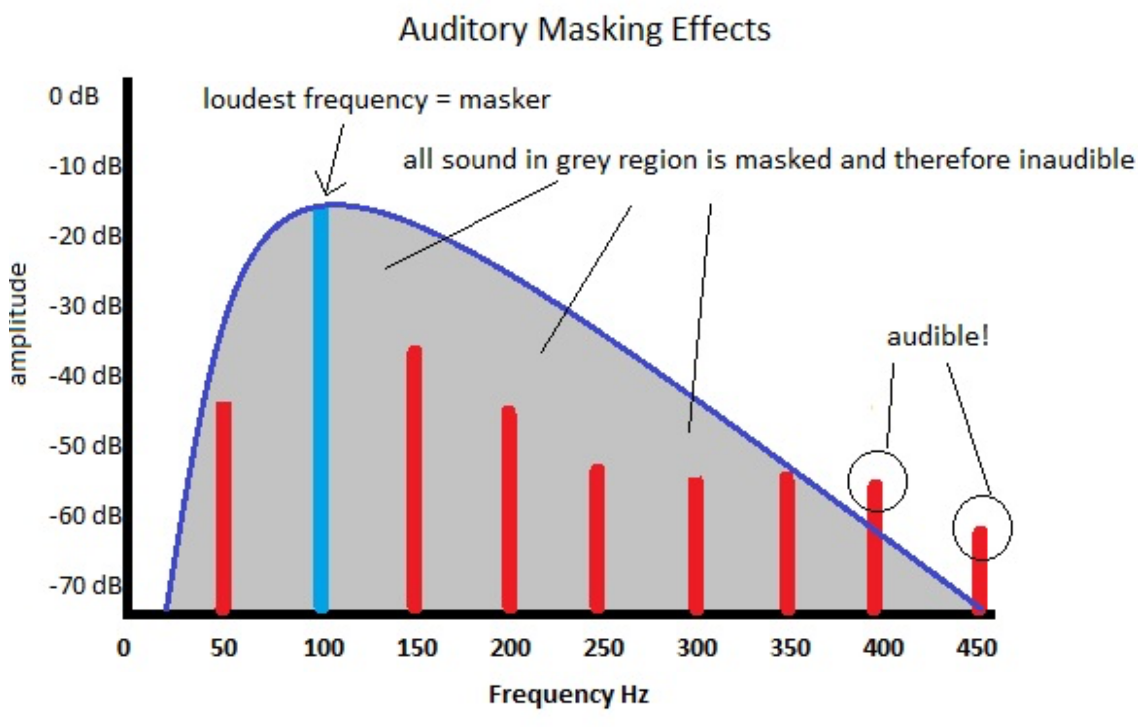

Figure 1. Auditory Masking in Frequency Domain. *(Image retrieved from www.audioholics.com)*

The effect of masking has been studied and many measurements have been proposed to quantify this phenomenon. In this thesis I am going to use the metric proposed by Aichinger and Sontacchi [7], called the Masked-to-Unmasked Ratio (MUR). The formula to compute this metric is presented in equation 3 in section 4.2.

3.4 Real-Time Performance On A Computer

A Real-Time performance on a Computer is mostly understood as a single performer using software and hardware to generate sounds while manipulating timbres using audio mixers, effects, and MIDI controllers. This allows performers to assume a role that borrows practices from conductors, mixing engineers, DJs, and instrumentalists [24]. The concept of live-sequencing exists and has been explored since the 1940s and constitutes the norm amongst live electronic music performers.

For this performance to be strictly considered Real-Time, it should guarantee response in defined time constraints and without perceivable delay. During the performance the output signal must be processed continuously and the time needed to process each sample must be shorter than the sampling period.

3.5 Basic Equaliser Theory

The broad description of an equaliser explains its functions in the sense of a device allowing to emphasise or attenuate the volume of specified frequencies. During the mix, equalisation is effectively used in different ways to correct problems that were created during the recording session or from incompatibility among instruments. Equalisation is used in a creative way in order to produce original effects [26].

The parametric equaliser is the most powerful and flexible of the equaliser types., it allows the mixing engineer or operator to emphasise or attenuate an arbitrary location in the audio spectrum.

An equaliser of a one third octave is a common design. This set up means each octave of the spectrum contains three bands, for example, starting at 1000 Hz, the following frequencies to be attenuated or emphasised will be: 1260 Hz, 1587 Hz, 2000 Hz, etc [27]. The number of bands is determined by their spacing and the requirement to cover the entire audible spectrum. Octave graphic equalisers usually have 10 bands, ranging from about 31 Hz at the lowest to 16 kHz at the highest. Third-octave designs usually have 31 bands ranging from 25 Hz to 20 kHz. These frequencies are standardised by the ISO [13]. This type of equaliser is often found in hardware units, because the gain sliders are easy to control. Rarely this kind of equalisers are used on individual channels for mixing, because those are far too inaccurate for a typical mix where the settings remain static during the whole recording.

Note that individual bands in an equaliser will not only amplify the range for the frequency it is labeled, for example 100 Hz - 1kHz. It will affect the frequencies around it as well.

Using an equaliser to unmask a signal with a fixed setting will decrease the dynamic impact of the masker, specially in cases where the masker and masked interact in a strong rhythmic way. In this case, a form of dynamic equaliser will allow the masker to retain its full tonal balance when the masked is absent. This is where modern equalisers implemented as plugins or computational tools are important to avoid killing musicality in the recording.

3.6 Spectral Processing and FFT Equalisers

Spectral Processing or sound processing in the frequency domain is a very important technique to achieve sound synthesis, transformations and processing in Real-Time situations. Real-Time spectral processing is relatively new in the world of music computing. Languages like Max/MSP or Pure Data have made it possible along with new developments in computer processing. This tools are widely used by artists, composers and researchers to process in Real-Time. Both enable work in the spectral domain via FFT analysis/re-synthesis [28].

Audio signals are composed of a multitude of different sound components. To process audio of a given signal, it is decomposed into building blocks that are better accessible. In the case these building blocks consist of complex-valued sinusoidal functions, such a process is also called Fourier analysis. The Fourier transform maps a time-dependent signal to a frequency-dependent function which reveals the spectrum of frequency components composing the original signal. Short Time Fourier Transform (STFT) allow us to compute the frequency-based representation of audio by determining the sinusoidal magnitude and phase content of local sections of a signal as it changes over time. In this way, the STFT does not only tell which frequencies are “contained” in the signal but also at which points of times these frequencies appear. The discrete Fourier transform (DFT), which can be efficiently computed using the fast Fourier transform (FFT), yields a discrete set of Fourier coefficients indexed by time and frequency parameters and is used to compute signals that are not perfectly periodic. The correct physical interpretation of these parameters in terms of units such as seconds and Hertz depends on the sampling rate, the window size, and the hop size used in the STFT computation.

Figure 2 shows a brief example of a signal transformed from time-domain to frequency-domain by taking small parts of time signal, and processing each one to compute the frequency information per window. During each window is possible to compute values, and descriptors in the frequency domain, as well as modifying this information to reconstruct a new version of the signal.

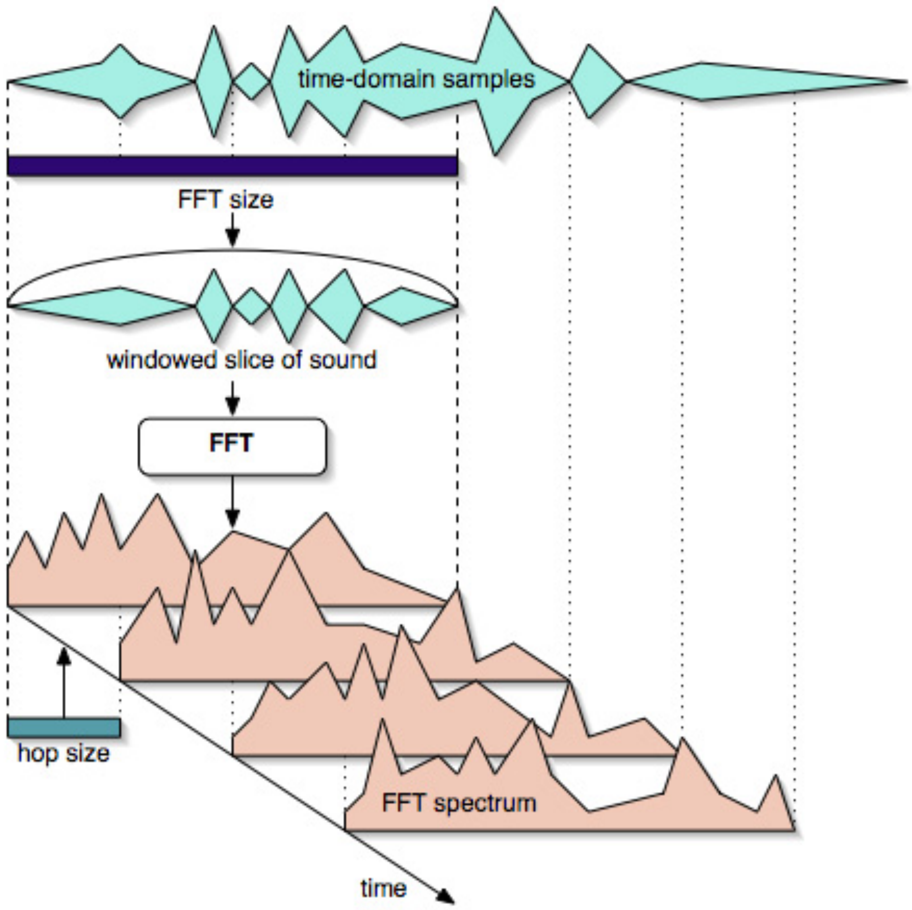

Figure 2. Diagram of the Short Term Fourier Transform (STFT). *(Image retrieved from https://cycling74.com/2006/11/02/the-phase-vocoder-—-part-i/)*

The STFT of a signal consists of the Fourier transform of overlapping windowed blocks of the signal. Windowed means applying a function of each block of information so that only certain section of the signal is nonzero. The shape of the window turns out to be important because the windowing produces a result where each bin of the transform (each component in the Fourier series), includes some energy from other bins nearby. The window function has the same result as a filter impulse response in frequency bins.

The FFT Size determines the size, in audio samples, of the overlapped windows which are transformed to and from the spectral domain. The window size must be a power of 2. The hop size (number of samples between each successive FFT window) is equal to the FFT Size divided by an overlap factor (e.g. if the frame size is 512 and the overlap is set to 4 then the hop size is 128 samples). The value must be a power of 2 and defaults to 2. A value of 4 is recommended for most applications.

In this thesis the process of computing the frequency domain of a signal using STFT is used to create an FFT Equaliser. This device has no mechanical or electronic counterpart. It can be viewed as a graphic equaliser with thousands of bands or simply as equalisation by drawing curves. In any case, it provides a detailed control of frequency response. There is an important difference between FFT and more traditional equalisers: FFT curves are often shown with equal frequency spacing across the window, whereas graphic equalisers have equal octave spacing [31]. For lower frequencies, the FFT equaliser presents a characteristic called "poor bass resolution". One way to compensate for this, is to use a large FFT Size.

Graphic equalisers work in time domain and FFT equalisers in frequency domain.

This thesis will use spectral processing with Max/MSP because it performs all the computation and transformations to relate time domain continuous musical signals into frequency domain, allowing for Real-Time windowing, manipulation of partials, frequency bins, amplitude in a multithreading environment inside a DAW, in this case Ableton Live.

3.7 Adaptive Digital Audio Effects

The implementation of the tools in this thesis follows the definition of Adaptive Digital Audio Effects (A-DAFX). This audio processes are effects driven by parameters or features extracted from the sound itself [23]. The principle of this effect's class is to provide a changing control adapting to dynamic signals as opposed to DAFx class, which is static.

A-DAFX sound processors combine the theory of sound transformation and adaptive control. This kind of processes are fundamental for real-time mixing and equalisation because the tools in this thesis apply different levels and configurations of effects based on the dynamic characteristics of the input signals. The interest in the development of tools like this is to provide possibilities for re-interpreting a musical sentences, changing timbres or presence of musical events with, strong, immediate or fine changes in the properties of the sounds [23].

3.8 Multithread Programming

In a multithread programming, a system is expressed as a collection of concurrent tasks. Using concurrent tasks rather than using a single task for all has advantages, like support for separation of tasks where each is designed, implemented, tested and verified. Another advantage is having concurrent tasks, where many processes run at the same time instead of a single sequential task [29].

Music and digital audio are a parallel structure, creating a set of non-communicating streams of tasks called voices, lines, or tracks. Most DAWs are organised around this concept. Examples include the way Ableton Live can use multiple cores and the multi-threading option in Max/MSP poly~ abstraction [30].

This thesis will use multithread programming to implement audio processing tasks like Short Time Fourier Transform in Real-Time using Max/MSP and Ableton Live.

3.9 Analysis Synthesis Techniques

In recent years a number of Digital Signal Processing algorithms have been developed to take an input signal and produce an output signal that is either identical to the input or a modified version of it. This process are based on different implementations of FFT and the input signal is assumed to be well represented by a model or mathematical formula with time-varying parameters, and the synthesis is simply the output of the model. The benefits of this techniques rely in the derived values for this analysis. The parameters can be modified to change the perceptual significance and musical utility of the result.

The FFT used for this techniques models the input as a sum of sine waves and the parameters to be determined by analysis are the time-varying amplitude and frequency for each sine wave. These sine waves are not required to be harmonically related, so this model is appropriate for a wide variety of musical signals. While this method work well for harmonic sounds, other percussive sounds (e.g., clicks) and certain signal-plus-noise sound combinations are not well represented by a model that sums a series of sine waves. These sounds can still be synthesised, but attempts to modify them can produce unexpected results [32].

This thesis will use Analysis Synthesis Techniques to decompose audio signals, modify the components and produce a new output adapted to the goals of the mix process. This processes will be detailed in section 5.

4 State Of The Art

4.1 Intelligent Music Production

Music production has been democratised in recent years, allowing anyone who has access to a computer to produce professional results. Musicians, singers and producers can perform the whole production process “in the box”. However, professional audio engineering knowledge is needed during this tasks. Access to professional tools at home, bedroom or small studio doesn’t guarantee professional results [1].

4.1.1 Intelligent Music Production vs. Human Engineers

Many products in the market today employ forms of intelligence used exclusively by humans in the past to solve actual music production problems. Besides some rare exceptions, most of these products do not try to fully automate sound engineering tasks or replace humans involved in the music making process. This products take advantage of computational power and precision to make the process of music making faster, fun and forward thinking. Human skills like passion, inspiration, communication are no competition for the perfect algorithm [2].

A human mixing engineer creating a mix would need to have control of multiple fader gains simultaneously. When masking is present, the mixing process becomes an iterative optimisation problem. With an automated process, the control of the fader gains, or interaction with the unmasking system offers advantages over the human process [25].

Amateur, or even professional artists can take advantage of Intelligent Music Production tools to concentrate on the artistic and creative process creating more expressive performances thanks to advancements in technology. Exploring the power and flexibility of computers can make the producers expression venture away from the rigid structure of the traditional toolset of music production [2].

4.2 Automatic Mixing

Automatic Mixing of multitrack music remains an unsolved problem that hasn't exhausted its research directions. Pioneers in this field believe this applications are still in its infancy [1].

The term "Automatic Mixing" appeared thanks to Dan Dougan, who invented a device that automatically reduces the strength of a microphone's while it is not receiving any signal. This device, the "automixer", was created for settings where no sound operator is present, in order to avoid feedback and noise in the signal. Dan Dugan showed his first "Adaptive Threshold Automatic Microphone Mixing System" in 1974 at the 49th Audio Engineering Society (AES) [14].

Between 2007 and 2010, Enrique Perez Gonzalez proposed automatic methods to adjust level, stereo panning, unmasking and delay correction, giving birth to the field of Automatic Mixing [1].

Further research has generated different proposals for automatic multitrack music mixing. The solutions proposed fall into this categories: Adjusting levels, panning, EQ, Compression and reverb. It is important to mention that not all proposals focus on real-time mixing.

In the market some tools are available for this task like: Neutron by Izotope, Vocal Rider by Waves, and Smart EQ by Sonible. From this tools, only the latter performs real-time processing after previous offline training.

4.2.1 Audio Engineering Best Practices For Unmasking

The research results in [15] compare three of the most commonly used techniques used by mixing experts to unmask sounds. Mirrored Equalisation, Frequency Spectrum Sharing, and Stereo Panning.

- **Mirrored Equalisation**: To unmask one channel in a specific frequency, the frequency is boosted on the masker signal and at the same time, reduced on the masker. The research by [5] suggests this approach may not work because it is more reliable to attenuate the masked frequency regions instead of boosting the unmasked frequency regions. Subjective tests show little preference for this solution.

- **Frequency Spectrum Sharing**: This approach uses the spectral centroid calculation for each track, and uses this information to determine which track is high pass filtered and which is low pass filtered. Results show listeners in the tests were indifferent of the application of this unmasking method.
- **Stereo Panning**: Adjusting pan position to reduce frequency masking proved to be the best method for unmasking. However, this solution may be unpractical because may be against artistic decisions, for example, we may not want to pan the kick drum or bass to one side of the recording.
- **Sidechaining**: This technique compensates the amplitudes of masker and masked signals by reducing the overall amplitude of the masker signal every time the masked reaches a certain threshold. This produces a highly dynamic manipulation of the masker signal that depends on the behaviour of the masked signal.

Many applications have been found for automatic mixing, from autonomous, to assistant and workflow enhancing tools. This solutions vary and have different degrees of user control. So far, solutions are divided in Real-Time and Offline

4.2 Previous Research

This section will refer and comment specific research made in the previous years that developed knowledge used as starting point in this thesis. The research publications in this section deal with unmasking of audio signals. This list will also help me to compare the solutions against each other, in order to find the most suitable for the case of Real-Time processing.

Perez-Gonzalez & Reiss 2008: Improved Control for Selective Minimisation of Masking Using Inter-channel Dependency Effects [16].

The author develops a tool that permits the enhancement of a source with respect to the rest of the mixture by selectivity unmasking its spectral content from spectrally related channels. It proposes one of the first documented masking metrics, the Spectral Masking (SM), computes the amount of overlap between the source and the rest of the mix. This measure is computed using the formula:

$$SM = (FFT(Ch_m))^2 - (FFT(mix - Ch_m))^2 \tag{1}$$

Where Ch_m represents the channel m, the one we want to measure masking against signal *mix*, which represent the sum of all channels. This metric depends on the size of the FFT used, and the author recommends not to use windowing. When $SM > 0$, the channel Ch_m is unmasked, otherwise is masked by the rest of the mix [16].

This research focuses on volume adjustments of channels and it implements a cross-adaptive channel enhancer performing a selective minimisation to enhance a user selected channel by ensuring it is spectrally unmasked from the rest of the mix.

Perez-Gonzalez & Reiss 2008: Automatic equalisation of multi-channel audio using cross-adaptive methods [17]. This paper is part of the series of papers published during the beginning of research in automatic mixing. The difference with the previous one lies in the fact that it proposes a method to apply equalisation of channels using five channels equalisers. Tests for this approach report inaccurate results on low frequencies and future work suggest the use of more equalisation bands.

Vega & Janer 2010: Quantifying Masking in Multi-Track Recordings [8]. In this research, the authors propose the Masking Coefficient or MC, a measurement of masking amount between two or more channels. This measure is a number between 0 and 1 and it corresponds to the amount of effective excitation overlap between two sounds. The masking coefficient between two excitations is given by the following equation:

$$MC = \frac{|e_1(t,b) - e_2(t,b)|}{60dB} \tag{2}$$

Where e_1 and e_2 are the matrices containing the excitograms of each sound in decibels.

The process of computing the *MC*, involves spectral processing and this may be a slow process causing latency and it's not intended for real-time tools.

Aichinger, et al. 2011: Describing the transparency of mixdowns: The Masked-to-Unmasked-Ratio (MUR) [7].

This masking measurement focuses on loudness and was tested perceptually through listening tests. It is computed as shown equation 3.

$$MUR = \frac{max(N_{masked}, 0.003)}{max(N_{unmasked}, 0.003)} \tag{3}$$

Where N_{masked} is the overall loudness of a signal when the masker is present, and $N_{unmasked}$ describes the overall total loudness of the same signal, when the masker is imaginary assumed not to be present. This metric is independent of frequency.

This metric is the most used and cited across most automatic mixing research reports, including the ones mentioned in this section.

Hafezi & Reiss 2015: Autonomous Multitrack Equalisation Based on Masking Reduction: In this research, the authors design a simplified measure of masking based on test practices in sound engineering. They implement offline and real-time versions of unmasking models [5].

Offline versions depend on history and tendencies of the mixed tracks. The equalisation setting remain constant over time and uses FFT to obtain the magnitude of each frequency region in a track.

In real-time version of the unmasking tool, the authors implemented an ISO standard octave-band (10 bands) processing on a frame-by-frame basis. Each filter or frequency band computes the RMS of the amplitude of the signal, which represents the magnitude of that band.

One of the most important conclusion of this research is that performance could have been improved by incorporating additional knowledge from psychoacoustics.

Robert Koria 2016: Real-Time Adaptive Audio Mixing System Using Inter-Spectral Dependencies [10].

This research proposes a method for real-time unmasking. The author describes and implements a model using Matlab, but just for test purposes. This implementation doesn’t run in real-time and is based on adaptive multi-band compressors analysing the spectral information between tracks.

To simplify the decision on which channel is the masked one, the author proposes a hierarchical cascade system where the sequence of the channels dictates the priority of attenuations applied (Figure 3). This sequence is defined by the user. The second

channel should be unmasked (or attenuated) from the first, the third channel should be unmasked from the second, and so on. This system helps to reduce the number of calculations during the process, making it more suitable for real-time situations.
Results from this research shows this attenuations occur mostly below 400 Hz.

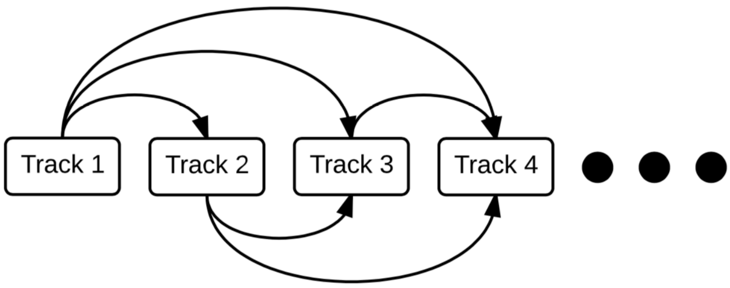

Figure 3. Track attenuation dependencies in a hierarchical cascade system [10].

De Man & Reiss 2017: Ten years of automatic mixing [1]. A literature review about the first ten years of automatic mixing. It sums up the most important milestones and the state of the art in this field. Cites the most important publications and research work during the period 2007-2017 and hints that Machine Learning will be the best approach for this kind of tasks.

In regards to this particular research, it doesn’t make a distinction between real-time or offline automatic mixing.

Ronan et al. 2018: Automatic Minimisation of Masking in Multitrack Audio using Subgroups [12]. Using subgroups is a technique employed by audio engineers, mostly the ones using digital tools, because it saves processing power and simplifies the task of processing sessions with large amount of channels. This research tests whether or not using subgroups is beneficial or not to automatic mixing systems.

The solution proposed in this research uses a compressor with static dynamic range for the entire track. The results were successful to reduce auditory masking between channels, but when compared against human mixes, the latter proved to create a better perception and clarity of the sounds involved in the mix.

5 Proposed Tool

In this thesis I will use the research from Chapter 4 to implement tools to help unmask the mix of two channels. Each implementation will use a different technique and will only unmask two channels (masker and masked). In further work, this tools can be extended to unmask a channel from mixes involving more channels. This tools will be compared in Chapter 8 to decide which technique is perceived as a better unmasking between two channels.

The tools developed in this research don’t intend to minimise the work of a professional audio engineer. As mentioned in section 4.1.1, this tool will work as an assistant to the audio engineer and will allow him to focus on the creative part of the mixing process. Also, this thesis is looking to develop tools for the live performer of electronic music on a computer, given this approach, this tool must be very easy to use and with and the interface should avoid complexity during the performance. I will minimise the number of controls to let the user focus on the artistic performance.

5.1 The Challenges Of A Real-Time Mixing Tool

The Real-Time tools developed in this thesis perform dynamic processes on the input signals, and cannot use algorithmic or iterative methods to unmask. Also, it won’t be able to process time domain masking.

Many research efforts on the field of mixing multitrack recordings use Machine Learning, Deep Learning or Artificial Intelligence techniques to achieve the goal of a balanced and unmasked mix. This processes typically pre-process the whole recording, or a representative segment of it. Others, like the ones proposed in [16] and [17] perform adjustments after listening the recording for a considerable period of time. This are then, oriented to studio work.

In real-time tools, the time allowed to take decisions is limited, and depends on the solution or tools used, but has to remain almost instantaneous and avoid producing high latency. The research on [33] and [34] has studied the amount of latency allowed for musicians, but no information was found regarding electronic music performance, where the performer is interacting with a computer system with possible delay compensation from the system, buffering and pre recorded segments. The amount of latency depends on the performance but becomes critical if contains live input from physical or analog instruments, or microphones. So far, this thesis will apply for performances where all elements are contained inside the computer.

Latency is also influenced by the processing power of the computer, in a way that the CPU of the computer hard limits the processing power allowed for performance, and while the process of unmasking the signal can be executed, it needs to permit the calculations needed for the sound generating elements, audio processes, MIDI and operating system tasks.

5.2 Platform Used

Max For Live running inside Ableton Live is the platform that I am using for this research and development. It provides the tools to analyse Real-Time information about this channel, implement band-pass equalisers and STFT, as well as to create and design the interface to allow the performer to interact with it.

Ableton Live is a complete DAW that can be used to mix, process and record live inputs like voice or instruments, to run virtual instruments and audio processes in Real-Time or as part of a live performance. This characteristics will permit to run unmasking tools in a real world scenario.

Version 8 of Max now permits us to implement the reception of all channels involved in just one patch, which makes it very practical to reduce complexity and configuration of the tools implemented. Previous research had to develop a plugin for each channel in the mix.

5.3 Restrictions Of This Implementation

As mentioned before, and for testing purposes, the first version of the unmasking tools will process two signals, the masker and the masked. This configuration will give immediate results necessary to come with conclusions about its efficiency. Listening test are conducted and are analysed in chapter 8.

It will be necessary to have the input of a performer or audio engineer to get professional results using this tools, because this will only focus on unmasking, not in the whole mixing process. If some other processes are needed like equalisation, change in amplitude, compression or distortion, it will be left to the user.

5.4 Implementation

For the purpose of testing different findings in the process of this development I created three different Intelligent Mixer devices (IM). This tools differ in the way the unmasking is implemented. The tools are named IM-WS (Whole Spectrum), IM-EQ (Equaliser) and IM-SEQ (Spectrum Equaliser). All implementations receive two signals, the masker and the masked signal as input and produce a single output signal, the processed mix of given channels. Figure 4 describes the signal flow and processes involved in the unmasking of two signals used by all implementations of IM devices.

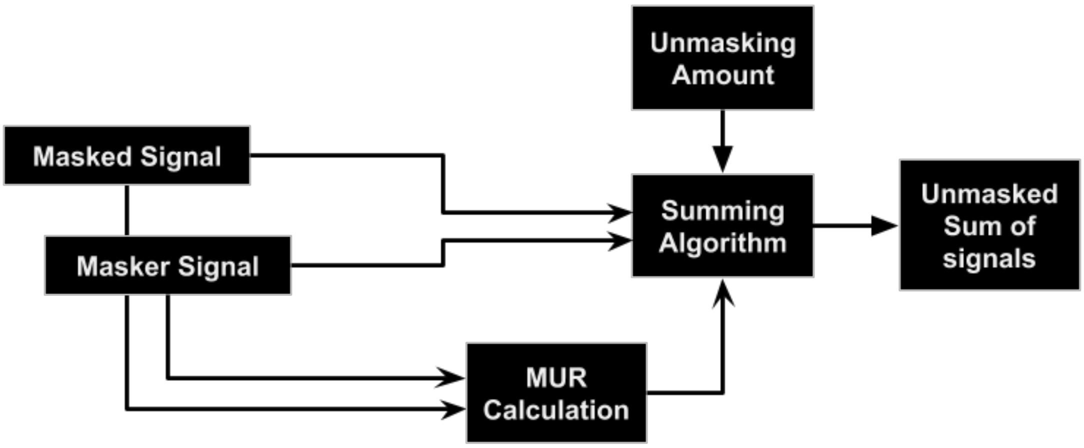

Figure 4. Block diagram of the unmasking process.

5.4.1 Implementation of IM-WS

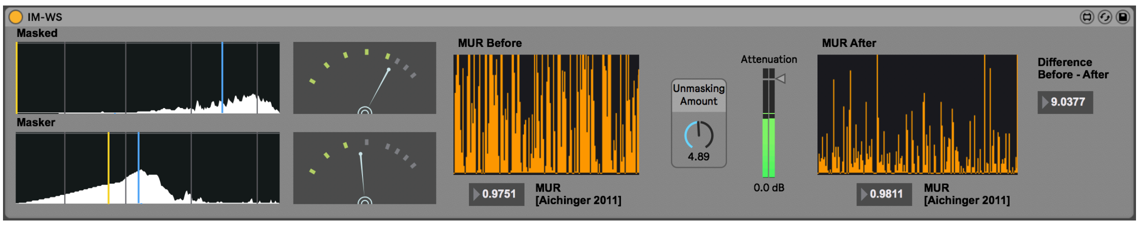

Figure 5. User interface of IM-WS.

This version of the unmasking tool computes MUR between masked and masked after short intervals of time. For the tests it was adjusted at intervals of 23 milliseconds, but this value is subjective and can be adjusted for CPU efficiency. At the moment, this parameter is not part of the user interface because goes beyond the scope of this thesis. Figure 5 shows the interface of IM-WS, where the Attenuation slider shows the real-time attenuation of the masker signal.

The only parameter needing user interaction is Unmasking Amount. This will let the user decide how much the unmasking process should affect the output signal, and consequently the mix of the channels involved.

This implementation performs unmasking by using the method described in [16], where the authors apply full range magnitude adjustments instead of equalisation techniques. In this case, the masker channel is attenuated in an amount given by the formula in equation 4.

$$A_t = UA_t * MUR_t \tag{4}$$

Where *A* is the amount of attenuation at time *t*, *UA* is the Unmasking Amount set by the user (in this implementation limited from 0 to 10) and *MUR* is the calculated ratio at time *t*.

This unmasking tool works in a similar way as a side chain compressor with fast attack and release, where the ratio is given by the Unmasking Amount parameter.

5.4.2 Implementation of IM-EQ

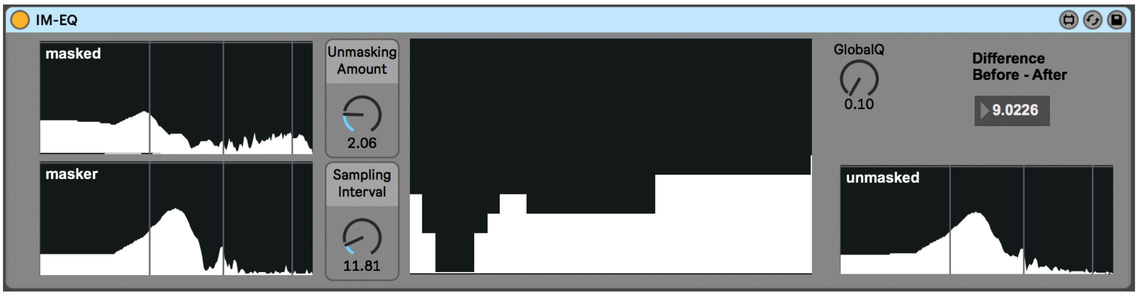

Figure 6. User interface of IM-EQ.

The research made by [17], uses a 5 band equaliser, since results were not quite as expected, it concludes that future work could use a higher number of equaliser bands. IM-EQ uses an ISO standard 31 band equaliser on the masker signal to create space in its spectrum allowing to hear the masked signal. The ISO 31 band equaliser [13], is a standard tool used widely in music production to shape the spectral envelope of an audio signal.

In this implementation, each signal, the masked and the masker is analysed at each of the 31 bands in separate processes, then MUR is computed per band to apply attenuation in the same way as IM-WS at each frequency band of the unmasking equaliser. This means 31 unmasking processes are running at the same time. Figure 6 shows the interface of IM-EQ, where the middle window shows 31 bars representing the the attenuation at each band of masker signal.

Having 31 band independent processes impacts the percentage of CPU used for this task, it is the equivalent to 31 times IM-WS. Also this device doesn't work frame by frame, it reads the values of each band and then waits for the number of milliseconds specified in the Sample Interval control to read values at each band again. The time between readings is used to compute MUR at each band and to apply the corresponding attenuations. This control can be used to balance between accuracy and CPU usage. At longer interval times, less CPU usage, a small value of this parameter means more detailed unmasking.

This implementation may lack accuracy when reading, computing MUR and applying attenuations, because the system doesn't provide all filter values at the same time. Also computing MUR and applying attenuations may not happen at the same time on every filter. Another important fact is that at each time interval, the attenuation applied corresponds to the MUR computed in the previous interval, causing unexpected results for large values of Sampling Interval. During tests, short intervals (no more than 50 ms) have produced accurate results, and proves the research made in [33] and [34]. Further research may be needed to find the maximum interval to produce accurate results, but this research goes beyond the scope of this thesis.

5.4.3 Implementation of IM-EQ

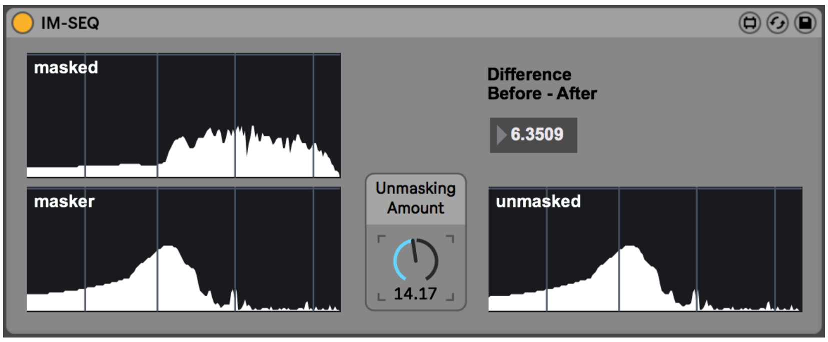

Figure 6. User interface of IM-SEQ.

The last tool implements a spectrum equaliser with an FFT size of 4096. The advantage of using spectral processing is that it is easy to implement in Max for Live and each of the 2048 bins of this implementation can run as separate parallel threads at the same time. Also, this implementation analyses the incoming audio in the frequency domain, where we are comparing spectral information directly. Each thread runs its own process of computing MUR and can attenuate itself according to the masking amount set by the user. This implementation is created to have more detail, synchrony and control in the unmasking process.

This version of the tool doesn't use 2048 bins for equalisation. From the work of Robert Koria [10], we have learned that frequencies above 400Hz don't need a great amount of attenuation when unmasking, thus, in order to simplify the process and make it less CPU intensive, I am using the first 38 bins. The center frequency for bin 38 is 409 Hz, which makes it enough for unmasking frequencies in the range mentioned before. Usually, in music, low frequencies contain the most energetic part of a recording.

When this device is running, it creates a concurrent set of 38 threads, where each one analyses its corresponding bin from the masker and masked signals. Once it has read the amplitude values of masker and masker, computes MUR and attenuation for the given bin and this attenuation is applied to the corresponding amplitude of the masker bin, all during the same FFT window. The process inside each bin is the same as the one described in section 6.1 for IM-WS.

Given the characteristics of FFT computing and resynthesis, this process can compute and apply attenuations to the data in the same window. This means the attenuation applied corresponds to the data read, and no latency issues are created like the one mentioned in section 6.2 for IM-EQ.

This process uses spectral processing to unmask the masker signal and create a resynthesised version of the signal below 409 Hz. After computing MUR and its corresponding attenuations, in order to have the whole masker signal spectrum with attenuations included, it is necessary to apply the audio process described in equation 5.

$$Masker = Masker - FFT(Masker, 38) + FFT(MaskerAttenuated, 38) \quad (5)$$

Where *FFT(Masker, 38)* represents the resynthesised signal using the first 38 bins of the FFT representation of the original Masker signal. *FFT(Masker Attenuated, 38)* represents resynthesised signal using the first 38 bins of the masker signal after attenuations have been applied. At the end of this process, Masker contains the unmasked signal.

At the end of this process, the signal below 409 Hz is resynthesised using inverse FFT. The reconstructed signal (unmasked masker) is part resynthesised and part original, so it may not produce the exact representation of the masker sound. After many tests, most sounds kept their timbral characteristics without distortion or noticeable artifacts.

6 How to use the tools

The unmasking tools have similar configurations and in this chapter, I will explain how to properly configure this tools inside an Ableton Live session. All tools use the same settings and configurations unless specified.

6.1 Installation

This tools can be downloaded from https://github.com/jjsauma/msmc. I provide the tools as separate files, one tool per file. In order to use it, it's necessary to drag the file from your computer's file manager application (Mac: Finder / Windows: File Explorer) to a channel inside the Ableton Live session. Also, it is possible to use many instances of the tools and/or a combination of the three tools inside the same Live session. The output from a tool can be used as the input of the next to create complex flows in the mixing process.

6.2 Where to allocate IM-Tools

When performing mixing of a multitrack recording, as discussed in section 3.2, it is necessary to separate artistic decisions and settings from technical issues. The tools developed in this thesis are intended to be used in the technical part of a mix and should be placed after all artistic settings and audio processes. For example, if a drum channel uses equalisation, compression and distortion, the IM tool must be used after distortion. The unmasking tools have to be used after all mixing processes in the effect chain, including volume setting of each channel, this tools are Post Fader devices.

IM tools can be set anywhere inside an Ableton Live session, but it is recommended to place it into a dedicated channel. Figure 8 shows an example where IM-WS implementation is placed in channel named IM-WS. The channel must be configured as "In" in the Monitor options. In the same example the channel "Drums" is the masked signal, the "Bass" channel is the masker signal. Using routing options from Ableton Live, masked signal is sent to "track in" (or channels 1 and 2) of the IM device. The masked signal is sent using channels 3 and 4 in the Audio To settings of the channel.

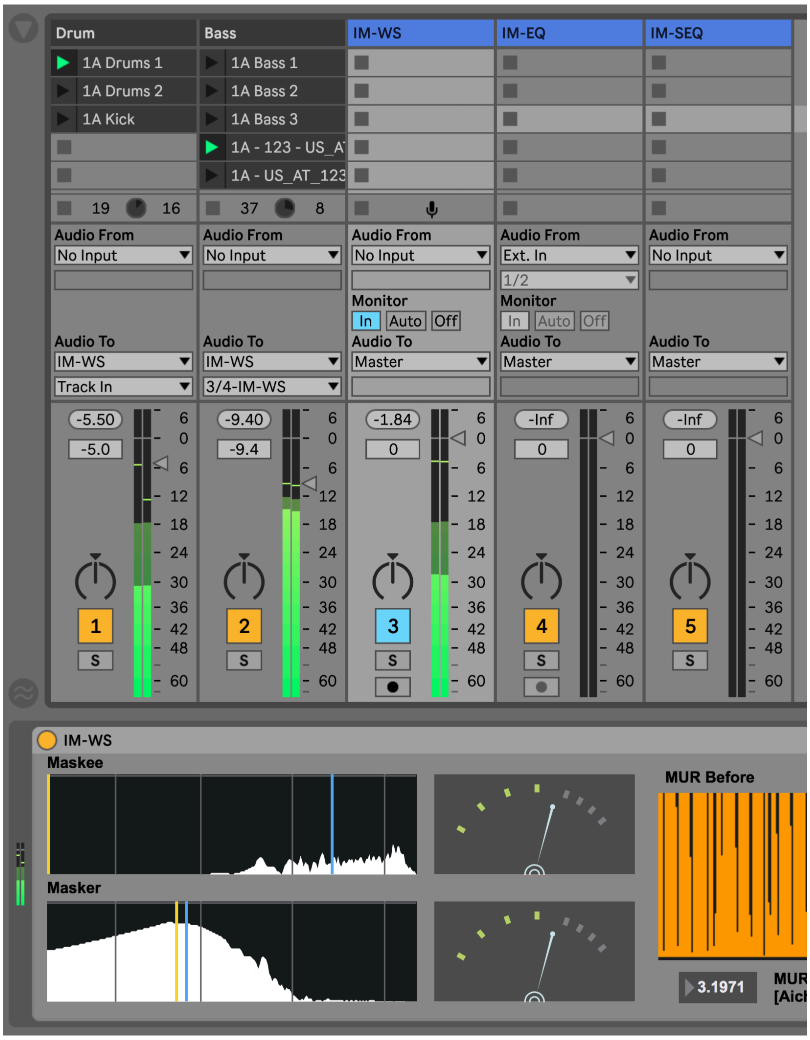

Figure 8. Where to place and how to route signals to IM Tools in an Ableton Live set.

6.3 Setting Parameters

Once the masker and masked signals are sent into the IM Tool inputs, the displays will show the input signals on corresponding spectrometer windows and a dial control named “Unmasking Amount”. When this control is set to zero (the minimum value), the device is not applying any unmasking algorithm to the signals.

Setting Unmasking Amount to values greater than zero will modify dynamically the masker signal to let the masked signal be heard. The right Unmask Amount control value is left to the user because it depends on taste. The Unmasking Amount won’t be the same value for each recording, because spectral characteristics of different recordings are not the same. Also the values among the different tools don’t imply the same effect of unmasking, because different techniques are applied.

6.4 How to process more channels

If you need to unmask from more than one channel, the process must be done grouping channels in pairs sorting the masking channels by priority as proposed by [10]. This process is adapted for this research and described in Figure 9.

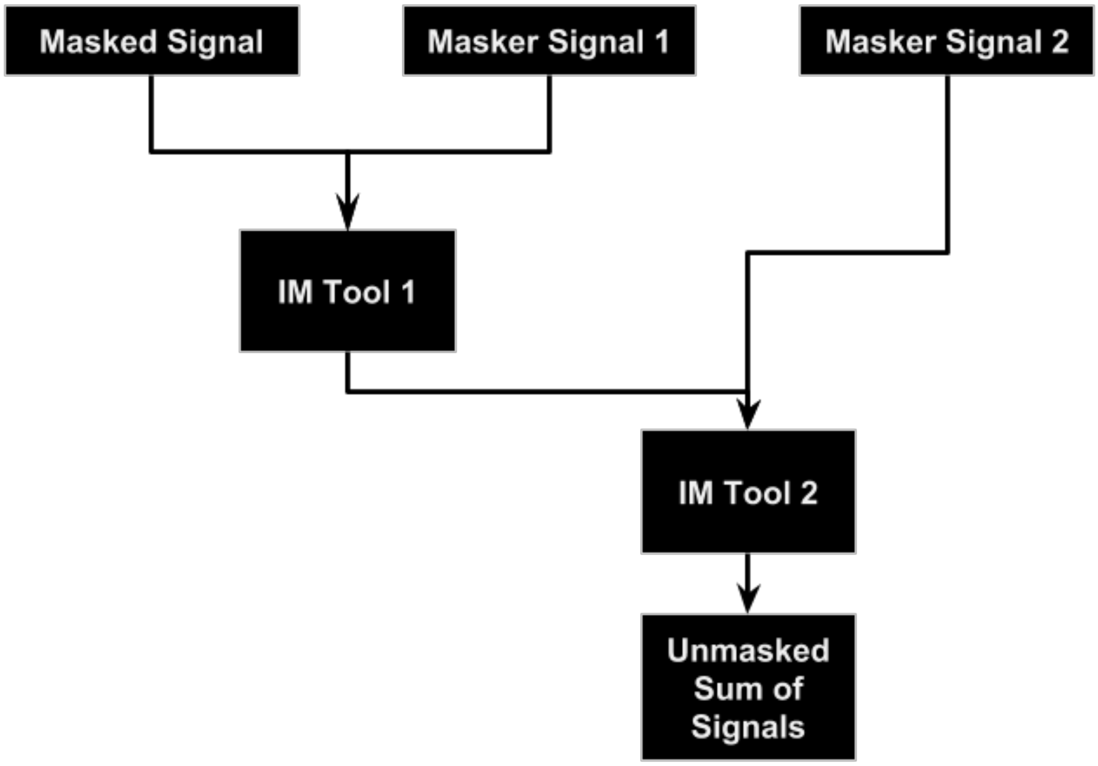

Figure 9. How to connect IM Tools to unmask from more than one channel.

7 Evaluation

To have a conclusion about the performance of the IM tools, it is necessary to conduct listening tests to find out which one is more adequate to reduce Auditory Masking in multitrack recordings.

Since I am measuring, and at the same time computing MUR, the objective comparison between the tools is evident and easy to compute using this metric. One the other side, a subjective evaluation is necessary to find how each tool affects perception.

7.1 Objective Evaluation

Through this research, I am using MUR to measure masking quantitatively inside each IM Tool. MUR has proven to be an accurate measurement [1], and the effect is measured by computing the difference between MUR before the unmasking process, and MUR after the process. A comparison between the unmasking tools using this number may be difficult to interpret, since the tools produce different results implementing different methods, and it's not a case of "more" or "less" unmasking measured with a number. The inherent measure for auditory masking, by definition, is perception. An objective evaluation of the outcome of IM Tools won't necessarily be accurate. For each tool, if we need to change the perceived unmasking we just need to change the Unmasking Amount value.

7.2 Subjective Evaluation

The final goal of this implementation and research is to create tools to allow the user to distinguish all sounds present in a mix. To have an opinion about how well the unmaking process works, I ran listening tests to measure perception of human listeners.

Each user is presented with three pairs of sounds, in this case a drum loop in one channel (the masked) and a bass sound in the other (the masker). Each pair is processed three times (one per tool). In total, each listener listens to 9 different mixes, grouped by the pair of sounds, named Mix 1, Mix 2, and Mix 3. After listening to the three different mixes for a pair of sounds, the user decides which mix sounds better and in his opinion, in which the sounds can be heard with more clarity and presence.

The tool with the more votes is the one performing better unmasking according to human perception.

To create this tests, the tools were adjusted at different values of Unmasking Amount. Each Unmasking Amount control was set to produce the same value of the difference between MUR before and after. In this way, the different implementations can be considered equivalent.

7.3 Results

Listening tests were conducted online and gathered a total of 40 participants where they had to select the IM tool they think produces a better unmasked version of each of three mixes. Figure 10 shows the chart of results for this test. The results clearly confirm preference for IM-EQ, the method using 31 independent and dynamic equaliser bands. The second preferred method is IM-SEQ, which unmasks sounds using spectral processing. The least preferred method is IM-WS, using just one compressor to unmask a signal.

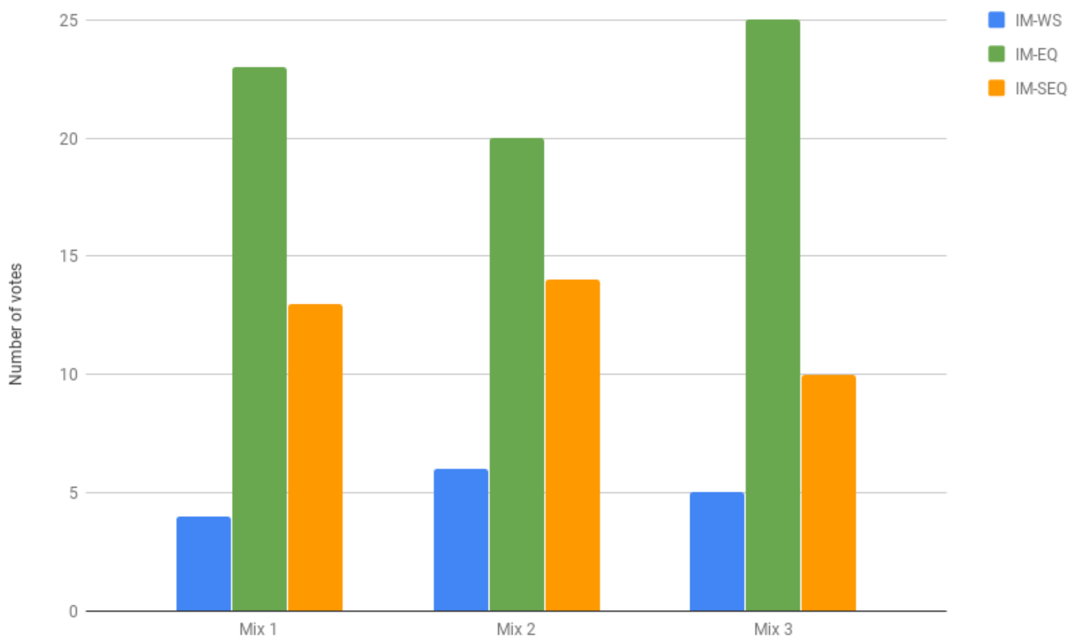

Figure 10. Results of listening tests. This chart shows how many times an IM Tool was marked as better masking algorithm for each mix.

Listeners who took this survey perceived the IM-EQ tool as the most natural sounding. Maybe this is a sign of the mixes we are used to hear. IM-SEQ was perceived as “too perfect” but effective. It would be interesting to test perception without listening to the unmasked tracks first, if comparing the sounds before and after unmasking doesn't create the feeling of an artificial mix, IM-SEQ could be a great option for tracks that are difficult to unmask.

CPU usage is an important factor when deciding which tool to use in a real world situation. Tests were run on a MacBook Pro 2017 computer with an 2.9 GHz Intel Core i7 processor.

IM-WS produces de the less CPU usage, and the rest of the tools produce results depending on their configuration. IM-EQ depends on Sampling Interval selected. During tests, this tool used almost 50% of the available CPU using the minimum Sampling Interval. For higher values, this process was almost not noticeable. IM-SEQ uses around 10% of the available CPU. No settings can be changed in this tool, it is possible to reduce the FFT size used to also reduce de CPU usage, but with a corresponding tradeoff in sound quality.

8 Conclusions and Future Work

8.1 Summary and Contribution

After a bibliographical review of intelligent mixing and auditory unmasking research (Chapter 4), a question about the feasibility of creating a real-time unmasking device for mixing was proposed, along with a research and documentation of the considerations for implementing such device. [1] Reflects on this research but concludes no definitive solution has been proposed.

Previous research in the field led to create 3 auditory unmasking devices implemented on Max for Live and running on Ableton Live with the intention of testing and using it on real time-performance of electronic music on a computer, and still leaving CPU time available for music performance. Section 4.2 describes the contribution of previous research to this thesis and tools and how this results will help in development and theoretical framework.

Chapter 6 describes the development of this devices, and documents to gain insights on the creation of real-time A-DAFX tools, along with equaliser theory for dynamic effects and spectral processing using real-time FFT computation.

In section 5.4.3 I propose and document a new method for auditory unmasking using spectral processing for real-time applications (which is usually CPU intensive). This method deals with the high processing percentage of other spectral implementations.

Listening tests were run to test human perception of the unmasking tools (Chapter 8). The results show that the 31 band equaliser implementation (IM-EQ) is better perceived when used dynamically to unmask sounds in a mix. The second best result was achieved by the spectral implementation (IM-SEQ), but it is far from the results obtained with IM-EQ. When comparing IM-EQ and IM-SEQ for perception, we can conclude that an equaliser performs a more natural sounding for auditory unmasking, even when its behaviour is not completely precise and synchronous. IM-SEQ produces an extremely precise blend of the masked and masker signals into a new signal that may include resynthesised sounds making the perceived sound an artificial mix.

On the CPU usage aspect, IM-EQ can be less expensive, but needs user interaction and understanding of how the Sampling Interval parameter affects the mix. This calls for specialised knowledge of the performer, which I am trying to avoid when developing this tool. Comparing with IM-SEQ, results suggest that multithreading adds CPU usage to the solution. Applying equalisation with an FFT is therefore generally very inefficient compared to comparable time-domain filters. I can conclude that IM-EQ produces the better balance between CPU usage and perception.

Previous research was intended to create a method or solution for generic tasks, the research made in this thesis seems to prove that adding limitations in order to focus on a defined environment helps us create a device to produce a reasonable solution.

8.2 Future Work

8.2.3 Multitrack device

Current devices can successfully unmask two channels, a device for unmasking more channels is possible to be developed instead of the suggested option of using many instances of IM Tools. This development may be able to optimise processing when analysing many channels at the same time. It will take more tests to decide how many channels can be processed by the device at the same time and still leave CPU time available for musical performance.

8.2.4 Using auditory system-like frequency scales

Using Mel or Bark Scales because it models the human auditory system behaviour may seem the following step in the development of a new version of IM-EQ. The implementation in this thesis allow for an easy reprogramming of a multi band equaliser using this scales. A comparison between an ANSI S1.11 [31], Mel and Bark Scales would help in developing a more natural sounding device for this task. Also, it will be easy to add a user interface control to select the frequency scale according to the needs of the performer.

8.2.5 Finding optimal Sampling Interval level

The Sampling Interval parameter was included to help save CPU usage while processing. More research is needed to determine the optimal interval and the cases where fixed intervals can be used according to the signals being processed. Fast Sampling Interval will help to unmask rhythmic and dynamic signals, and a slow value will use less CPU when unmasking sustained and continuous sounds.This device can have profiles or use Machine Learning algorithms to find the best profile for the signals being processed.

8.2.6 Mixing and Mastering

Spectral processing provides an accurate mix of the two channels, and the real-time method developed here may be adapted and tested for purposes of Frequency Spectrum Target Mixes, which is a current trend in digital mastering systems and production techniques [35], [36].

8.2.7 DJ-style mix

IM-SEQ has proved to be a blending tool that may have applications in DJ-style mixes, where the goal is to create transparent blend of songs one after another. IM-SEQ blends two signals in a way that big amplitude jumps and sudden changes are avoided by continuous analysis of their frequency spectrum.

8.3 Reproducibility

IM-Tools developed and tested during this thesis, as well as surveys and data, are stored in https://github.com/jjsauma/msmc. It includes Excel spreadsheets, Max for Live Patches and Ableton Live Set used for this research.

9 Bibliography

1. De Man, B., Reiss, J. Ten years of automatic mixing. Queen Mary University of London (2017).
2. Kokkinis, E. Artificial Intelligence in Music Making: is a Jobless Future Ahead of Us? (2017). Retrieved from: http://www.avidblogs.com/artificial-intelligence-music-making-jobless-future-ahead-us/
3. Moore, B. An introduction to the psychology of hearing. Academic Press, 6th Edition (2013).
4. Hermes, K. Towards measuring music mix quality: the factors contributing to the spectral clarity of single sounds. University of Surrey (2017).
5. Hafezi, S., Reiss, J. Autonomous Multitrack Equalisation Based on Masking Reduction. Centre for Digital Music. Queen Mary University of London, London, UK (2015).
6. Martinez Ramirez, M., Reiss, J. Stem Audio Mixing as a Content-Based Transformation of Audio Features. Centre for Digital Music. Queen Mary University of London, London, UK (2017).
7. Aichinger, P., Sontacchi A., Schneider-Stickler B. Describing the Transparency of Mixdowns: The Masked-to-Unmasked-Ratio. 130th Audio Engineering Society Convention (2011).
8. Vega, S., Janer, J. Quantifying Masking in Multi-Track Recordings. Universitat Pompeu Fabra (2010).
9. Puckette, M., Apel, T., Zicarelli, D. Real-time audio analysis tools for Pd and MSP. UCSD, Cycling '74 (1998).
10. Koria, R. Real-Time Adaptive Audio Mixing System Using Inter-Spectral Dependencies. Linköping University. (2016).
11. D. Dugan. Automatic Microphone Mixing. J. Audio Eng. Soc., vol. 23, June 1975.
12. Ronan, D. Ma, Z. Nanamanra, P. Gunes, H. Reiss. J. Automatic Minimisation of Masking in Multitrack Audio using Subgroups (2018).
13. ANSI S1.11: Specification for Octave, Half-Octave, and Third Octave Band Filter Sets. American National Standards Institute (2004).
14. United States Patent 3,992,584. (1976).

15. Wakefield, J., Dewey C. An Investigation into the efficacy of methods commonly employed by mix engineers to reduce frequency masking in the mixing of multitrack musical recordings. University of Huddersfield (2015).
16. Perez-Gonzalez, E., Reiss, J. Improved Control Of Selective Minimisation of Masking Using Inter-Channel Dependency Effects. Centre for Digital Music, Queen Mary University of London (2008).
17. Perez-Gonzalez, E., Reiss, J. Automatic Equalisation of Multi-Channel Audio Using Cross-Adaptive Methods. Centre for Digital Music, Queen Mary University of London (2008).
18. Vega, S., Janer J. Quantifying Masking In Multi-Track Recordings. Universitat Pompeu Fabra (2010).
19. Gibson, D. The Art Of Mixing. A Visual Guide to Recording Engineering, and Production. Mix Books. 2005.
20. Owsinski, B. The Mixing Engineer’s Handbook 4th Edition. Bobby Owsinski Media Group. 2017.
21. White, P. Mixing Essentials. Sound On Sound Magazine (2006). Retrieved from: www.soundonsound.com/techniques/mixing-essentials.
22. Ma, Z. Intelligent Tools for Multitrack Frequency and Dynamics Processing. Queen Mary University of London (2016).
23. Verfaille, V., Arfib, D. A-DAFX: Adaptive Digital Audio Effects. CNRS - LMA (2001).
24. Baytas, M., Göksun, T., Özcan, O. The Perception of Live-sequenced Electronic Music via Hearing and Sight. NIME’16 (2016).
25. Terrell, M., Simson, A., Sandler, M. The Mathematics of Mixing. Queen Mary University of London (2014).
26. Principles of Equalization (2013). Retrieved from: www.izotope.com/en/blog/mixing/principles-of-equalization.html
27. Välimaki, Vesa., Reiss, J. All About Audio Equalisation: Solutions and Frontiers. Department of Signal Processing and Acoustics, Aalto University (2016).
28. Charles, J. A Tutorial On Special Sound Processing Using Max/MSP and Jitter. Computer Music Journal. MIT (2008).
29. Muller, H. Designing Multithreaded And Multicore Audio Systems. XMOX Ltd. Bristol, UK (2011).
30. Real-Time Musical Applications on an Experimental Operating System for Multi-Core Processors. University of California, Berkeley (2011).
31. Elsea, P. The Art And Technique of Electroacoustic Music. A-R Editions (2013).
32. Dolson, D. The Phase Vocoder: A Tutorial. Computer Music Journal, Vol 10, No. 4. (Winter, 1986).

33. Schuett, N. The Effects of Latency on Ensemble Performance. Stanford University (2002).
34. C. Bartlette, D. Headlam, M. Bocko, and G. Velikic. Effect of Network Latency on Interactive Musical Performance. Music Perception. Volume 24, Issue 1 (2006).
35. Elowsson, A., Friberg, A. Long-term Average Spectrum in Popular Music and its Relation to the Level of the Percussion. KTH Royal Institute of Technology, School of Computer Science and Communication, Speech, Music and Hearing. 2017.
36. Duarte Pestana, P., Ma, Z., Reiss, D., Barbosa, A. Black, D. Spectral Characteristics of Popular Commercial Recordings 1950-2010. Catholic University of Oporto, Queen Mary University of London, 2013.